Effective *k*-Means Clustering in Greedy Prepruned Tree-based Classification for Obstructive Sleep Apnea

Doreen Y. Y. Sim[1], Ahmad I. Ismail[2], and C. S. Teh[1]
[1] Department of Cognitive Sciences, Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak, Kota Samarahan, Malaysia
[2] Department of Respiratory Medicine, UiTM Medical Specialist Centre, Faculty of Medicine, Universiti Teknologi MARA, Selangor, Malaysia
Email: dsdoreenyy@gmail.com; csteh@unimas.my; izuanuddin@uitm.edu.my

***Abstract*—Incorporation of prepruned decision trees to *k*-means clustering through one to three types of tree-depth controllers and cluster partitioning was done to develop a combined algorithm named as Greedy Pre-pruned Tree-based Clustering (GPrTC) algorithm. Pre-pruned clustered decision trees are applied in a greedy concerted way to five datasets of obstructive sleep apnea and others from online data repositories. The optimal number of *k* clusters for *k*-means clustering is determined after trees are greedily prepruned by tree-depth controllers of minimum number of leaf nodes, minimum number of parent nodes and maximum number of tree splitting. After applying the GPrTC algorithm to the assigned datasets, when compared with the conventional *k*-means clustering, results showed that the former has significantly lower average distortion per point and lower average run-time for 2-D and 3-D data over around 30 thousand points. Classification efficiency and speed of the former algorithm is more than two times better the latter algorithm over a higher range of points being run. GPrTC algorithm showed better classification accuracies than *k*-means clustering in almost all the assigned datasets. This concludes that the proposed algorithm is significantly much more efficient, less distortion and much faster than *k*-means clustering with moderately better in terms of classification and/or prediction accuracies.**

***Index Terms*—Pre-pruned decision trees, *k*-means clustering, tree-depth controllers, GPrTC algorithm, average distortion per point, average run-time**

I. Introduction

K-means clustering is a method of vector quantization [1]-[8] which its main drawbacks are proposed to be minimized or compromised when it is pre-processed by greedy classification through pre-pruned decision trees.

A. Greedy Tree-Based k-Means Clustering

Both decision trees (DTs) and *k*-means clustering can be considered greedy and relatively weak classifiers. *k*-means clustering is an unsupervised machine learning algorithm which partitions data into *k* number of mutually exclusive clusters [9]-[12]. To make these two classifiers working in tandem to achieve synergistic effects in this research, decision trees will be pre-pruned if the characteristics of the datasets applied are having instances which can be pre-pruned by the tree-depth controllers. While performing conventional classification, *k*-means clustering has a major limitation that user has to initially specify the number of clusters, i.e. *k*, manually. Decision tree or DT, however, does not suffer from this major drawback [13]-[17].

B. Greedy Tandem in Greedy Pre-pruned Clustering

Why considered as a greedy tandem? DT is a non-parametric and unstable classifier which can be pre-processed by *k*-means clustering algorithms in order to achieve better classification results [9]-[12]. The "first favorable, first serve basis" of decision trees and *k*-means clustering technique are considered to be the 'greedy' algorithms. This logic is 'hidden' from the explicit views and principal component knowledge through visualization when this combined algorithm is to perform optimal classification and find optimal choices [2], [13]-[17]. In other words, the aforementioned Greedy Pre-pruned Tree-based Clustering, i.e. GPrTC, algorithm never reconsiders its choices and/or future choices, but conveniently chooses the apparently or visually most beneficial choice on a "whatever the most favorable, whatever will be chosen" basis. So, GPrTC algorithm is a greedy algorithm.

Table I shows five datasets of Obstructive Sleep Apnea (OSA) actual patients' records (each marked with * sign) and four datasets acquired online from the UCI data repositories. The second column is the number of instances (i.e. tuples) of each dataset, the third column displays the number of Tree-Depth Controllers or TDCs conducting pre-pruning, while the last column shows the number of features which are not being pruned by TDCs before applying the clustering. For the datasets of Monks2 and Iris, since all features are being pre-pruned by the TDCs of 'MaxNumSplits', there is no feature not being pruned by TDCs before clustering.

Manuscript received October 27, 2021; revised December 29, 2021; accepted January 27, 2022.
Corresponding author: Doreen Y. Y. Sim (email: dsdoreenyy@gmail.com).

doi: 10.18178/ijeetc.11.3.242-248

TABLE I: DETAILS OF DATASETS APPLIED WITH GPRTC ALGORITHM

Assigned Datasets (from UCI data repositories and actual OSA patients' records)	Characteristics of the assigned datasets			
	No. of tuples	Number of features	Number of TDCs to conduct pre-pruning by greedy tree splitting	Number of features not being pruned by TDCs before clustering
OSA Dataset 1*	430	6	1	5
OSA Dataset 2*	450	8	2	5
OSA Dataset 3*	350	10	2	7
OSA Dataset 4*	390	12	1	11
OSA Dataset 5*	490	14	2	11
Monks2	432	6	1	0
Titanic	1309	4	2	1
Diabetes	768	8	2	5
Iris	150	4	1	0

In estimating the number of *k* clusters, each cluster formed by *k*-means clustering may become each of the tree splitting decisions for the decision trees to be constructed. Another limitation of *k*-means clustering is that it can only handle numerical data, and *k*-means clustering always assumes spherical clusters to be applied and that each cluster has about equal numbers of observations [3]-[8], [12]. All these limitations are proposed to be counter-reduced by preprocessing the *k*-means clustering with pre-pruned tree-based algorithms.

Datasets applied: nine datasets were applied using the proposed GPrTC algorithm and *k*-Means clustering. Four of the datasets were acquired online from the UCI data repositories. The rest of the datasets were acquired from the actual Obstructive Sleep Apnea (OSA) patients' records collected from the Sleep Lab of public hospitals and Neurology private clinics in Malaysia. Table I shows the details of the datasets that were applied to the GPrTC algorithms and *k*-means clustering. It shows the number of tuples (i.e. instances of each dataset), number of features of each dataset, number of Tree-Depth Controllers (TDCs) to conduct pre-pruning by greedy tree splitting before *k*-means clustering, and the number of features not being pre-pruned by TDCs before *k*-means clustering.

II. PROPOSED ALGORITHMS

A. Theoretical Background and Research Hypothesis

Most of the datasets applied are acquired online from the UCI data repositories and OSA actual patients' records. Since the conventional *k*-means clustering has certain major limitations or drawbacks, it is proposed to be "working in tandem" with pre-pruned decision trees. The main disadvantages of using *k*-means clustering are as follows: (1) in conventional *k*-means clustering, choosing the optimal *k* manually (i.e. the number of clusters) can be difficult especially when the clusters are of very different densities and sizes; (2) this classical approach can only handle numerical or continuous data, but not categorical data; (3) it assumes that each cluster has about the same number of observations since it presumes that users only deal with spherical clusters [5]-[8], [12].

Research Hypothesis: To overcome major drawbacks of *k*-means clustering that the number of clusters, i.e. *k*, must be specified beforehand, this research hypothesizes that by combining pre-pruned decision trees and *k*-means clustering while adopting the greedy selection fashion during the process, a synergetic effect will be achieved so as to reach the research aim of more efficient and/or more accurate classification to the assigned datasets. The optimally selected *k* value by the greedy approach is hypothesized to be relatively smaller than that obliviously selected by the classical *k*-means clustering.

Research Aim: Embarking on the highly concerted greedily selected solutions (i.e. "apparently" optimal and/or sub-optimal solutions which are opposite to the conventional approaches) during the pre-pruning process of the decision trees by one to three different types of tree-depth controllers, can significantly avoid the under-fitting classification, infeasible or inefficient effects which are usually caused by the oblivious selection of *k* clusters by the *k*-means clustering.

Research Objectives: 1) To show that optimization of the decision tree pre-pruning through greedy selections can overcome or minimize the major drawbacks and limitations of *k*-means clustering; 2) To prove that the classification and/or prediction efficiency can be significantly improved by incorporating *k*-means clustering in the greedy pre-pruned tree-based classification and/or prediction; 3) To illustrate and prove that the overfitting effects or drawbacks of greedy pre-pruning approaches can be significantly minimized while maintaining or improving the classification and/or prediction accuracies when being combined with *k*-means clustering.

Fig. 1 shows the greedy classification process of this research, i.e. pre-pruned decision tree-splitting and *k*-means clustering, of GPrTC algorithms. Why greedy classification is chosen in this research? Many problems in Machine Learning can be solved or partially solved by repeatedly doing whatsoever seems to be the best at the moment, i.e. without looking at the future or in a long-run holistic solving plan. By adopting the greedy classification fashion, GPrTC algorithm tries to 1) assign each point to the closest cluster center, 2) compute a nearby new cluster center as the centroid of each cluster, and 3) in *k*-means clustering, partition data into *k* number of clusters as much 'mutually exclusive or furthest apart' as possible, i.e. in a greedy fashion.

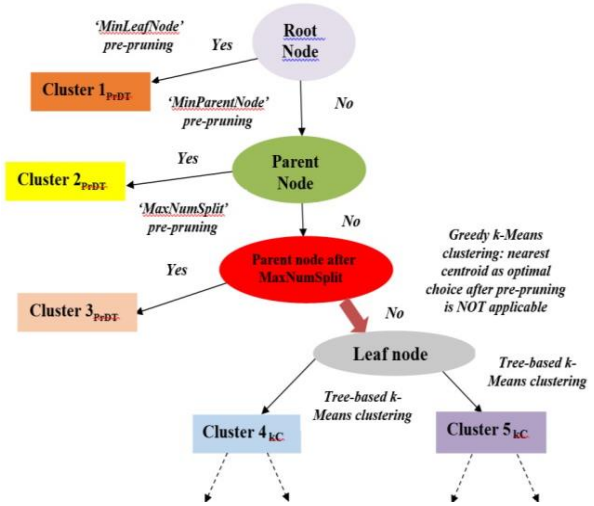

Fig. 1. Greedy classification algorithm using the pre-pruned tree-based *k*-means clustering, i.e. GPrTC algorithm.

Fig. 1 shows the procedures of GPrTC algorithm, i.e. greedy pre-pruning processes of decision tree-splitting was performed before conducting *k*-means clustering. Based on information gain theory for the decision trees to split until reaching pure nodes [7]-[8], [10]-[11], [13], one to three types of tree-depth controllers (TDCs) were performed based on the features and characteristics of the assigned datasets. These TDCs used greedy selections by selecting the optimal or sub-optimal values to reach the pure nodes of the decision trees through the very Minimum Leaf Node Size, Minimum Parent Node Size and the very Maximum Tree-Splitting Numbers. All these are based on the greedy classification fashion. As shown in Fig. 1, if the characteristics and features of the datasets are not eligible for greedy pre-pruning by TDCs, GPrTC algorithm will still be proceeded with greedy *k*-means clustering until the optimal classification is achieved.

In conventional approaches, to avoid decision tree overfitting with or without being combined with other classifiers, decision trees or DTs will choose the largest value of Minimum Leaf Size and/or Minimum Parent Size so that DTs will not have to split so many times. In the same vein, classical DTs will also choose the smallest value of Maximum Number of Tree-Splitting so that the trees will not have to split so many times in order to reach the pure nodes [7]-[9], [13]-[17].

In this research, the greedy decision tree pre-pruning approaches are based on the 'first favorable, first serve' features and characteristics of the datasets which can be the 'best feasible at the first sight' to three major types of Tree Depth Controllers, i.e. the 1) very minimum number of leaf nodes (MinLeafNode or MNL); 2) very minimum number of parent nodes (MinParentNode or MNP); and 3) very maximum number of tree splitting (MaxNumSplits or MNS).

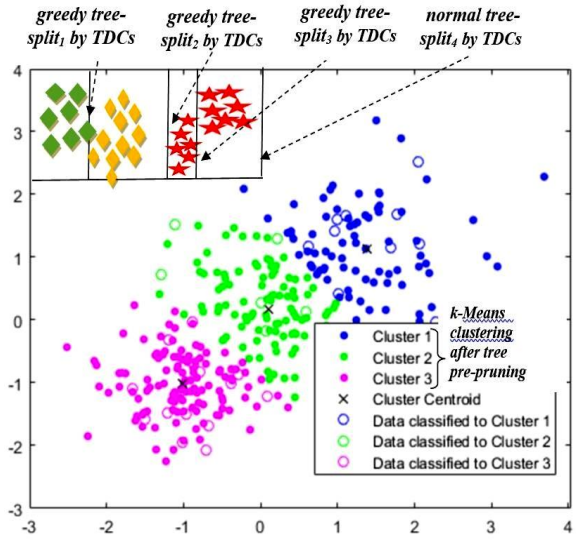

Fig. 2. Generalized classification outcome of cluster partitioning after running the GPrTC algorithm to the assigned datasets.

Fig. 2 shows one of the generalized classification outcomes of GPrTC algorithm after applying the cluster partitioning to the assigned datasets. It shows the first 3 greedy tree-splitting by Tree-Depth Controllers (TDCs) and another normal tree-splitting by TDCs before the *k*-means clustering is implemented. The reason that the first 3 tree-splits are considered as greedy tree-splits is because the data points are being 'forcefully' partitioned despite 'densely populated' surrounding the centroids. The 4th tree-split is considered as normal tree-split since the partition cut is done at one of the 'least populated' data points which are farthest from the centroids on both sides. Since the pre-pruning processes are based on Information Gain and greedy strategies, 'MinLeafNode' (MNL) and 'MinParentNode' (MNP) are respectively the very minimum number of leaf nodes and the very minimum number of parent nodes for a decision tree to reach the smallest pure node, while 'MaxNumSplits' (MNS) is the very maximum number of tree-splitting for DT to reach the smallest pure node.

B. Datasets Applied and Algorithmic Aspects

Table II shows how the proposed GPrTC algorithm adopts the greedy selection of the optimal values from the derived ranges by the three types of tree-depth controllers (i.e. Minimum Leaf Size, Minimum Parent Size and Maximum Number of Tree Splitting) while applying *k*-means clustering in each of the assigned datasets. The value highlighted in **bold** is the 'greedy' value selected among the range of values 'pruned' or 'trimmed' by the Tree-Depth Controllers (TDCs).

As shown in Table II, the last column is the greedy values, i.e. optimal and/or sub-optimal values selected based on the greedy pre-pruning approaches by the tree-depth controllers (TDCs).

TABLE II: GREEDY SELECTIONS OF GPRTC ALGORITHM BY THE TREE-DEPTH CONTROLLERS (TDCS) AND SUB-OPTIMAL SOLUTIONS

Assigned Datasets	Greedy selections by PrTC algorithm based on **optimal solutions** (TDCs)/ **suboptimal** choices			
	Tree-Depth Controllers (TDCs)			Greedy option selected
	Min Leaf Size	Min. Parent Size	Max. Num. Splitting	
OSA Dataset 1*	**8**-18	N/A	N/A	**8**
OSA Dataset 2*	**8**-18	**36**-50	N/A	**8**; **36**
OSA Dataset 3*	**6**-12	**36**-48	N/A	**6**; **36**
OSA Dataset 4*	**7**-14	N/A	N/A	**7**
OSA Dataset 5*	**12**-18	**67**-74	N/A	**12**; **67**
Monks2	N/A	N/A	41-**45**	**45**
Titanic	**6**-11	**36**-300	N/A	**6**; **36**
Diabetes	**7**-12	**45**-46	N/A	**7**; **45**
Iris	N/A	N/A	3-**45**	**45**

C. Implementation Aspects Using Assigned Datasets

The experiment and programming platform used is the latest version of MATLAB R2020b and Microsoft Excel of MS Office 2021.

D. Procedures of GPrTC Algorithms

The stepwise description on how *k*-means clustering works is as follows: Step i) initially, arbitrarily specify the number of clusters, *k*; Step ii) initialize cluster centroids (i.e. cluster centers) by iteratively shuffling the dataset and then randomly selecting data points for the centroids without replacement; and Step iii) the iteration is carried on until there is no change to the centroids (which means that the data points to the assigned

clusters are not changing anymore). In Step iii), the following equation:

$$F = \sum_{i=1}^{q} \sum_{k=1}^{K} P_{ik} \left\| x_i - \mu_k \right\|^2 \quad (1)$$

is used to a) calculate the sum of the squared distance in between the data points and all centroids; b) assign each of the data points x_i to the nearest cluster (i.e. centroid); and c) calculate the centroids for the clusters by taking the average of all the data points that belong to each cluster [1]-[6], [8], [10]-[12].

Equation (1) is the objective function of *k*-means clustering, where $P_{ik} = 1$ for the data point x_i if it belongs to the cluster *k*; while if otherwise, $P_{ik} = 0$. In this equation, μ_k is the centroid of x_i's cluster [8], [10]-[12].

The approach that *k*-means clustering works to solve the problem is known as the expectation-maximization approach. The E-step (i.e. to fulfill the expectation stage) assigns the data points to the closest cluster, while the M-step (i.e. to fulfill the maximization stage) computes the centroid of each cluster [8], [10]-[12].

As shown in (1), the minimization to the problem has two parts. Firstly, minimize *F* with respect to P_{ik}, but remain the μ_k as fixed. Secondly, minimize *F* with respect to μ_k, but remain P_{ik} as fixed. In terms of technical illustration, firstly differentiate *F* with respect to P_{ik} and then update the cluster assignments (i.e. E-step). Secondly differentiate *F* with respect to μ_k and re-compute the centroids after cluster assignments from previous step or E-step (i.e. M-step). In equation (1), the E-step is that we assign the data point x_i, to the nearest cluster which is judged by its sum of squared distance from the cluster's centroid. Then, the M-step is that we re-compute the centroid of each cluster to reflect the new assignments. Equation (1) is to minimize the pairwise squared deviations of points in the same cluster [8], [10]-[12].

As shown in Algorithms I and II below, |*D*| is the total number of tuples or instances, i.e. data records, in the dataset, i.e. *D*. GPrTC algorithm consists of two algorithms: greedy pre-Pruning algorithm and greedy *k*-Means Clustering algorithm.

Algorithm I: GreedyPrune
Al I: *gPrune*(*N*, *vmin_MNL*, *vmin_MNP*, *vmax_MNS*)
/* *gPrune* is greedy pruning, *N* stands for a node */
/* *vmin_MNL* is the very minimum number of 'MinLeafNode' */
/* *vmin_MNP* is the very minimum number of 'MinParentNode' */
/* *vmax_MNS* is the very maximum number of 'MaxNumSplits' */ **if** *N is a pureN* **then** *N.Stop* = TRUE
/**pureN* is a pure node */
else *gPrune*(*N*, *vmin_MNL*, *vmin_MNP*, *vmax_MNS*);
if *N.MNL* <= *vmin_MNL**|*D*|
then *N.MNL.Stop* = TRUE
else *gPrune*(*N*, *vmin_MNL*, *vmin_MNP*, *vmax_MNS*);
end
if *N.MNP* <= *vmin_MNP**|*D*|
then *N.MNP.Stop* = TRUE
else *gPrune*(*N*, *vmin_MNL*, *vmin_MNP*, *vmax_MNS*);
end
if *N.MNS* >= *vmax_MNS**|*D*|
then *N.MNS.Stop* = TRUE
else *gkMeansC*(*CL*, *vmin_MSE*); /*
proceed to Algorithm II: greedykMeansC */
end

Algorithm II: GreedykMeansC
Al II: *gkMeans*(*CL*, *vmin_MSE*)
/* *gkMeans* is greedy k-Means clustering */
/* *CL* stands for clusters */
/* *vmin_MSE* is the very minimum Mean-Squared Errors
for the distance between the member distance and the cluster center */
if *CL.MSE* <= *vmin_MSE**|*D*|
then *CL.MSE.Stop* = TRUE
else
CL.MSE.Stop = FALSE
end

E. Implementation Results, Comparisons and Contrasts

Tables III, IV and V show mainly the implementation results and outcomes of GPrTC algorithm and *k*-means clustering over a significant range of points.

Table III is a detailed comparison and contrast of the average distortion per point over a significant range of data points between the GPrTC algorithm and *k*-means clustering. GPrTC algorithm consistently shows much lower average distortion per point when compared with *k*-means clustering.

To elucidate Table III in picture illustration form, Fig. 3 illustrates the trends of improvements for each of GPrTC algorithm and *k*-means clustering in graph forms for the average distortion per point over a significant range of 50k data points. Although *k*-means clustering shows less average distortion per point when data points increase to greater values, the improvement in efficiency is still quite a long way to catch up with the former.

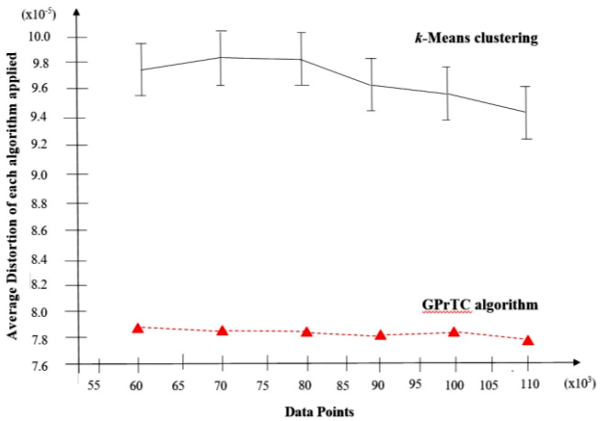

Fig. 3. Generic performance for 2-D and 3-D data of the assigned datasets showing the average distortion per point for k-Means clustering and GPrTC algorithm over an average range of 35 iterations and 30 runs.

TABLE III: COMPARISONS OF THE AVERAGE DISTORTION PER POINT BETWEEN GPRTC ALGORITHM AND K-MEANS CLUSTERING

Data points ($\times 10^3$) during average distortion measure	Classification efficiencies of algorithms applied			
	Average distortion per point ($\times 10^{-5}$)			Difference (%) in efficiency between two algorithms
	GPrTC algorithm	*k*-means clustering	Differences in between two algorithms	
60	7.86	9.74	1.88	23.92
70	7.83	9.81	1.98	25.29
80	7.81	9.80	1.99	25.48
90	7.80	9.60	1.80	23.08
100	7.82	9.57	1.75	22.38
110	7.76	9.40	1.64	21.13

Fig. 3 shows the generic performance of the distortion measurement per point of GPrTC algorithm and *k*-means clustering for the 2-D (2 dimensions) and 3-D (3 dimensions) data of the assigned datasets. It demonstrated that GPrTC algorithm has significantly much lower error rates or distortion per point if compared with the classical *k*-means clustering. This is because the former algorithm is able to minimize or overcome the main drawbacks of the latter, i.e. relatively poor scaling computational ability of *k*-means clustering, the initial number of *k* clusters has to be supplied by the user, and the search is prone to local minima. In addition, from Fig. 3, GPrTC algorithm has shown consistently low average distortion per point over a running range from 60k to 110k data points. On the contrary, *k*-means clustering has shown relatively high average distortion per point at the beginning but it improves by having its distortion per point reducing from around 9.7×10^{-5} to around 9.5×10^{-5} over the same data point range run by the GPrTC algorithm. Table IV is another detailed comparison and contrast of classification efficiency of GPrTC algorithm and *k*-means clustering but it is in terms of run time in seconds. The last column shows the percentage of increase or decrease from the starting point when the two algorithms are being run and recorded their average run-times.

Table IV shows the classification efficiencies in terms of the average run times of the two algorithms of GPrTC and *k*-means clustering over a significant range of points. GPrTC algorithm shows much shorter run time for each point per measure and this efficiency is maintained over a running range of 6k to 30k of points. In contrast, *k*-means clustering shows much slower run time and in the higher running range of data points, the run time is much longer over a higher range of point values.

TABLE IV: RUN TIME RESULTS (IN SECONDS) OF GPRTC ALGORITHM AND *K*-MEANS CLUSTERING OVER A SIGNIFICANT DATA RANGE

Data Points ($\times 10^3$) during average run time	Classification efficiencies of algorithms applied			
	Run time per measure (in seconds)			Percentage (%)in run time from the starting data point
	GPrTC algorithm	*k*-means clustering	Differences in between two algorithms	
6	31.0	98.0	67.0	0.00
8	59.0	127.0	68.0	1.49
10	73.5	147.0	73.5	9.70
12	75.5	177.0	101.5	51.49
14	87.5	190.0	102.5	52.99
16	93.0	213.5	120.5	79.85
18	102.0	251.0	149.0	122.39
20	119.0	256.0	137.0	104.48
22	105.0	277.0	172.0	156.72
24	104.0	280.0	176.0	162.69
26	128.5	306.5	178.0	165.67
28	136.0	307.5	171.5	155.97
30	141.0	348.5	207.5	209.70

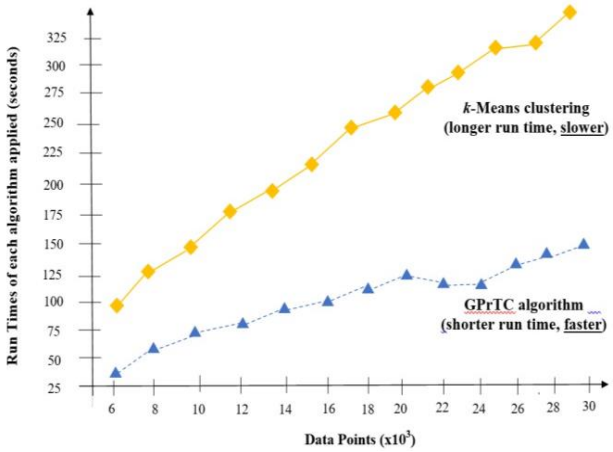

Fig. 4. Average run-times of *k*-Means clustering and GPrTC algorithm for data of 2-D and 3-D in the assigned datasets over a range of 30 thousand points in a period recorded in seconds.

TABLE V: CLASSIFICATION ACCURACIES OF GPRTC ALGORITHM AND *K*-MEANS CLUSTERING

Assigned Datasets	Classification accuracies of algorithms applied			
	Average classification accuracies and optimal *k* value selected			% of differences in between two algorithms
	k-means clustering algorithm	GPrTC algorithm	Optimal value of *k* selected	
OSA Dataset 1*	0.9395	0.9930	2	5.35
OSA Dataset 2*	0.8800	0.9467	4	6.67
OSA Dataset 3*	0.9114	0.9829	3	7.15
OSA Dataset 4*	0.8487	0.9256	3	7.69
OSA Dataset 5*	0.8327	0.9245	4	9.18
Monks2	0.8843	0.9144	2	3.01
Titanic	0.7693	0.7983	3	2.90
Diabetes	0.7096	0.7656	4	5.60
Iris	0.9467	1.0000	3	5.33

Fig. 4 is the illustration of the comparison and contrast of the speed, i.e. run time in seconds, in between GPrTC algorithm and *k*-means clustering in graph form. Fig. 4 shows a comparison and contrast of the average run-time of GPrTC algorithm and *k*-means clustering for data of 2D (2 dimensions) and 3-D (3 dimensions) in the assigned datasets over a range of around 30 thousand (i.e. 30×10^3) data points in a period recorded in seconds. It shows clearly that GPrTC algorithm has an average run-time of more than double the average run-time of *k*-Means clustering. In other words, GPrTC algorithm runs much faster (i.e. more than 2 times the speed for run-time per second), and hence much more efficient than *k*-Means clustering.

Table V shows the comparison and contrast of the classification accuracies in between GPrTC and *k*-means clustering algorithms, as well as optimal value of *k* selected for GPrTC algorithm and *k*-means clustering algorithm when being applied to each assigned dataset. Since the 3rd column is the optimal value of *k* clusters selected for the two algorithms, it is the value of *k* which can produce the highest average classification accuracy for each assigned dataset. Since GPrTC algorithm shows better classification accuracy, it can be seen that with greedy selection fashion, each optimal value of *k* selected is relatively of small value. This is tally with the greedy concerted way as suggested in the research

hypotheses. Since the clustered trees have had pre-pruned by the TDCs, the optimal *k* value selected while implementing GPrTC algorithm was shown (as in Table V) to be relatively smaller than the *k* value selected while applying the classical *k*-means clustering.

Fig. 5 illustrated clearly the generic version of Receiver Operating Characteristic (ROC) curves for both GPrTC algorithm and *k*-means clustering when being applied to the assigned datasets. Both these algorithms showed relatively good ROC performance, but in terms of classification and/or prediction abilities, GPrTC algorithm performs better than *k*-means clustering.

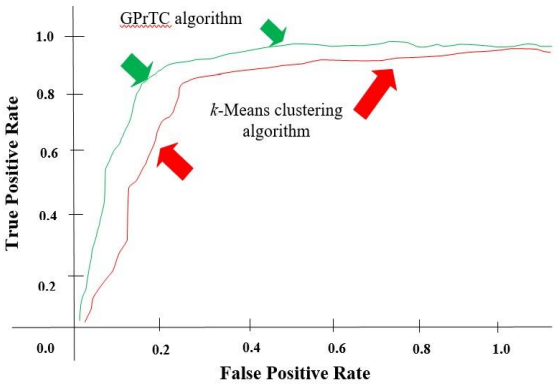

Fig. 5. Generic graph of Receiver Operating Characteristics (ROC) curves of GPrTC algorithm and *k*-means clustering applied to the assigned datasets.

III. Discussions

GPrTC algorithm not only shows much better classification efficiency than *k*-means clustering, but is also able to maintain its efficiency over a higher running range of points. In addition, in terms of average running time, GPrTC algorithm shows much lower run-time and is also able to improve its speed and efficacy in more than 2 times than *k*-Means clustering over a greater range of points. On the contrary, *k*-means clustering shows less classification efficiency and its less efficacy is getting more profound (i.e. more than 2 times of earlier classification inefficiencies) over a higher running range of points. Although the research aim and objectives are shown to be achieved, GPrTC algorithm may not be able to exert its strengths when it is being implemented in certain distribution of data points where the major drawbacks of *k*-means clustering cannot be overcome by greedy tree-based pre-pruning approaches, or when the tree-depth controllers applied in greedy fashion are infeasible during the classification and/or prediction.

For datasets which have features of quite uniform and/or of 'monotonous' distribution, i.e. the leaf node is quite difficult or very hard to achieve pure or almost pure node despite almost endless decision tree-splitting, GPrTC algorithm may not be able to show significantly better and more efficient classification and/or prediction results that *k*-Means clustering. In other words, GPrTC algorithm will perform much more efficient and much better classification and/or prediction results than *k*-means clustering when the distribution of data and the features of the datasets are 'tunable' and 'pre-prune-able' by tree-depth controllers in a greedy concerted way.

IV. Conclusion

The research aim and objectives have shown to be achieved from the implementation results and outcome above. In terms of classification and/or prediction efficiency when measuring the iteration cum running time, results of average distortion per point and classification accuracies, the proposed and implemented Greedy Prepruned Tree-based Clustering (GPrTC) algorithm shows significant better and more efficient classification results when being compared with the classical *k*-means clustering. It shows much lower average run-time and much less average distortion per point during the implementation. For classification accuracy to all the assigned datasets, when compared with *k*-means clustering, the proposed GPrTC algorithm also showed better outcome.

Conflict of Interest

The authors declared that there is no conflict of interest for the submitted work in this research.

Authors' Contributions

The first author analyzed data from the collected patients' records of Obstructive Sleep Apnea (OSA) and online data repositories, developed all algorithms and did the write-up. Second author conducted the research in OSA while the last author conducted the research for the datasets from online repositories. All authors agreed to the final version of this journal paper.

Acknowledgment

All the applied datasets, except those with indicated * signs (which were collected from the actual patients' records of Obstructive Sleep Apnea (OSA) in the Sleep Labs of the public hospitals and Neurology clinics in Selangor, Malaysia), were online acquired from the University of California Irvine (UCI) data repositories.

References

[1] A. M. Ikotun, M. S. Almutari, and A. E. Ezugwu, "K-means-based nature-inspired metaheuristic algorithms for automatic data clustering problems: Recent advances and future directions," *Appl. Sci.*, vol. 11, no. 23, 11246, pp. 1–61, Nov. 2021.

[2] R. Ananda and A. Prasetiadi, "Hierarchical and k-means clustering in the line drawing data shape using procrustes analysis," *Int. J. Inform. Visualization*, vol. 5, no.3, pp. 306–312, Sep. 2021.

[3] D. Dai, Y. Ma, and M. Zhao, "Analysis of big data job requirements based on *k*-means text clustering in China," *PLoS One*, vol. 16, no. 8, pp. 1–6, Aug. 2021.

[4] P. Sreelatha, J. F. Banu, T. Ch. A. Kumar, D. Sugumar, S. K. Rawat, and A. J. Niazi, "Improved clustering using deep learning model on water resource engineering," *Biosc. Biotech. Res. Comm.*, vol. 14, no. 6, pp. 343–349, Jul. 2021.

[5] M. Akyol, "Clustering hotels and analyzing the importance of their features by machine learning techniques," *J. Comp. Sci.*

Technol., vol. 2, no. 1, pp. 16–23, Jun. 2021.

[6] R. Vankayalapati, K. B. Ghutugade, R. Vannapuram, and B. P. S. Prasanna, "*K*-means algorithm for clustering of learners performance levels using machine learning techniques," *Revue d'Intelligence Artificielle*, vol. 35, no. 1, pp. 99–104, Feb. 2021.

[7] D. Y. Y. Sim, C. S. Teh, and A. I. Ismail, "Correlation feature selection weighting algorithms for better support vector classification: An empirical study," *Solid State Technol.*, vol. 63, no. 2, pp. 2794–2805, Oct. 2020.

[8] D. Y. Y. Sim, "Extensive incorporation of *k*-nearest neighbor to support vector machine through correlation studies for a better classification," *Test Eng. Management*, vol. 82, pp. 11898–11907, Jan. 2020.

[9] D. Y. Y. Sim, "Support vector machine pre-pruning approaches on decision trees for better classification," in *Proc. 2th Int. Conf. Electronics and Electrical Eng. Technol.*, Malaysia, 2019, pp. 30–36.

[10] S. Kanjanawattana, "A novel outlier detection applied to an adaptive *k*-means," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 5, pp. 569–574, Oct. 2019.

[11] N. Rachapudi, L. Ganesh, A. Sekar, *et al.*, "Discovery of structured data using unsupervised spatial clustering and human supervision," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 5, pp. 586–591, Oct. 2019.

[12] D. Y. Y. Sim, "Redefining the white-box of *k*-nearest neighbor support vector machine for better classification," *Lecture Notes in Electrical Engineering*, vol. 603, pp. 157–167, Aug. 2019.

[13] D. Y. Y. Sim, C. S. Teh, and A. I. Ismail, "Pushing constraints by rule-driven pruning techniques in non-uniform minimum support for predicting obstructive sleep apnea," *Appl. Mech. Mater.*, vol. 892, pp. 210–218, June 2019.

[14] D. Y. Y. Sim, C. S. Teh, and A. I. Ismail, "Pushing visualization effects into pushed schema enumerated tree-based support constraints," *Appl. Mech. Mater.*, vol. 892, pp. 219–227, June 2019.

[15] X. Qiao, J. Bao, H. Zhang, F. Wan, and D. Li, "Underwater sea cucumber identification based on principal component analysis and support vector machine," *Measurement*, vol. 133, pp. 444–455, Jan. 2019.

[16] D. Y. Y. Sim, C. S. Teh, and A. I. Ismail, "Improved boosted decision tree algorithms by adaptive apriori and post-pruning for predicting obstructive sleep apnea," *Adv. Sci. Lett.*, vol. 24, no. 3, pp. 1680 –1685, Jan. 2018.

[17] D. Y. Y. Sim, C. S. Teh, and A. I. Ismail, "Improved boosting algorithms by pre-pruning and associative rule mining on decision trees for predicting obstructive sleep apnea," *Adv. Sci. Lett.*, vol. 23, no. 11, pp. 11593–11598, Aug. 2017.

Doreen Y. Y. Sim acquired her Doctor of Philosophy (Ph.D.) in Computational Intelligence, Data Mining and Machine Learning in the year 2018 after she graduated with M.Sc. and B.Sc. (Honors) degrees in Business Information Technology respectively from University of Portsmouth and University of Central England in Birmingham, United Kingdom. She also has a double major Medical Sciences degree which she acquired from University of Otago, Dunedin, New Zealand. She has extensive, i.e. more than one and one third of a decade, of fulltime lecturing experience in Data Mining, Machine Learning and Computational Intelligence. She has around 13 years of research experience in the same field as well as in Artificial Intelligence. She is currently working as a Research Fellow, Educator cum Data Scientist in the educational and healthcare research industries ever since she worked as a full-time Computing, IS cum BIT Lecturer and Research Fellow in the private and public universities in Malaysia for many years. She has around 14 recent publications, with certain research articles published in high impact factor ISI-indexed Tier-1 and SCOPUS-indexed international journals as well as other research papers in peer review international conference proceedings.

Dr. Sim has been a very active research cum conference committee member, technical committee member, paper reviewer, sessional chairperson, and invited speaker in various international conferences, symposiums and seminars. She is also an invited paper reviewer in certain ISI-indexed Tier-1 international journals such as Information Sciences. She has been an active research committee member in a few professional committees such as being a CBEES committee member in Hong Kong Chemical, Biological and Environmental Engineering society. Two of her recent publications were awarded with the 'Best Paper Awards' and another one of her recent publications was awarded with the 'Best Presentation Award'.

Ahmad I. Ismail is a Medical Specialist in Respiratory Medicine at UiTM Medical Specialist Centre, and an Associate Professor in the Faculty of Medicine, Universiti Teknologi MARA, Selangor, Malaysia.

C. S. Teh is a Ph.D. holder in Virtual Reality and Artificial Intelligence. He is an Associate Professor in the Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak, Malaysia.